findPC: An R package to automatically select number of principal components in single-cell analysis

Haotian Zhuang

Introduction

findPC is a software tool including six methods to automatically determine the optimal number of principal components to retain based on the standard deviations explained by each PC. A major advantage of findPC is that the only information required is a series of standard deviations explained by each PC.

Installation

findPC software can be installed via Github. Users should have R installed on their computer before installing findPC. R can be downloaded here: http://www.r-project.org/. To install the latest version of findPC package via Github, run following commands in R:

```
if (!require("devtools"))
install.packages("devtools")
devtools::install_github("haotian-zhuang/findPC")
library(findPC)
```

findPC function

The synopsis of findPC is:

```
findPC(sdev,number = 20,method = 'perpendicular line',aggregate = NULL,figure = FALSE)
```

The default is to return the optimal number of PCs by Perpendicular line with 20 PCs. The following codes take the 50 PCs of human fetal brain tissue as an example.

```
sdev<-c(6.909905,5.891180,5.110216,3.996021,3.541697,
        2.708781,2.636483,2.476862,2.420636,2.345929,
        2.285914,2.197029,2.157942,2.067661,1.977431,
        1.905260,1.882831,1.819860,1.797364,1.763118,
        1.734229,1.721318,1.700343,1.694309,1.682183,
        1.677859,1.668176,1.658513,1.656305,1.648881,
        1.629452,1.626978,1.622602,1.614844,1.610602,
        1.603669,1.598074,1.593364,1.586666,1.584247,
        1.580981,1.574226,1.568660,1.567932,1.562935,
        1.557284,1.554781,1.551753,1.547596,1.543069)
findPC(sdev = sdev)
```

```
##       Perpendicular line
## 20PCs                  6
```

The argument 'sdev' should be sorted in decreasing order.

```
findPC(sdev = -sdev)
```

```
## Error in findPC(sdev = -sdev): 'sdev' should be sorted in decreasing order
```

Number

The argument 'number' is a vector including number of PCs used in the following function.

```
findPC(sdev = sdev,number = 51)
```

```
## Error in findPC(sdev = sdev, number = 51): 'number' exceeds the available number of PCs
```

```
findPC(sdev = sdev,number = c(16,20,28))
```

```
##       Perpendicular line
## 16PCs                  6
## 20PCs                  6
## 28PCs                  6
```

Method

The argument 'method' specifies the six methods or returns the six results simultaneously.

```
findPC(sdev = sdev,method = 'xxx')
```

```
## Error in findPC(sdev = sdev, method = "xxx"): 'method' includes 'all','piecewise linear model',
##     'first derivative','second derivative','preceding residual',
##     'perpendicular line (default)','k-means clustering' options
```

```
findPC(sdev = sdev,number = c(16,20,28),method = 'all')
```

```
##       Piecewise linear model First derivative Second derivative
## 16PCs                      6                6                 6
## 20PCs                      6                6                 6
## 28PCs                      6                6                 6
##       Preceding residual Perpendicular line K-means clustering
## 16PCs                  4                  6                  4
## 20PCs                  6                  6                  4
## 28PCs                  6                  6                  5
```

Method 1: Piecewise linear model

```
findPC(sdev = sdev,number = c(16,20,28),method = 'piecewise linear model')
```

```
##       Piecewise linear model
## 16PCs                      6
## 20PCs                      6
## 28PCs                      6
```

Method 2: First derivative

```
findPC(sdev = sdev,number = c(16,20,28),method = 'first derivative')
```

```
##       First derivative
## 16PCs                6
## 20PCs                6
## 28PCs                6
```

Method 3: Second derivative

```
findPC(sdev = sdev,number = c(16,20,28),method = 'second derivative')
```

```
##       Second derivative
## 16PCs                 6
## 20PCs                 6
## 28PCs                 6
```

Method 4: Preceding residual

```
findPC(sdev = sdev,number = c(16,20,28),method = 'preceding residual')
```

```
##       Preceding residual
## 16PCs                  4
## 20PCs                  6
## 28PCs                  6
```

Method 5: Perpendicular line

```
findPC(sdev = sdev,number = c(16,20,28),method = 'perpendicular line')
```

```
##       Perpendicular line
## 16PCs                  6
## 20PCs                  6
## 28PCs                  6
```

Method 6: K-means clustering

```
findPC(sdev = sdev,number = c(16,20,28),method = 'k-means clustering')
```

```
##       K-means clustering
## 16PCs                  4
## 20PCs                  4
## 28PCs                  5
```

Aggregate

If users are also interested in the mean, median, or voting (median if all are different, otherwise mode) of the result, the argument 'aggregate' will support them.

```
findPC(sdev = sdev,number = c(16,20,28),method = 'all',aggregate = 'mean')
```

```
## mean
##    6
```

```
findPC(sdev = sdev,number = c(16,20,28),method = 'all',aggregate = 'median')
```

```
## median
##      6
```

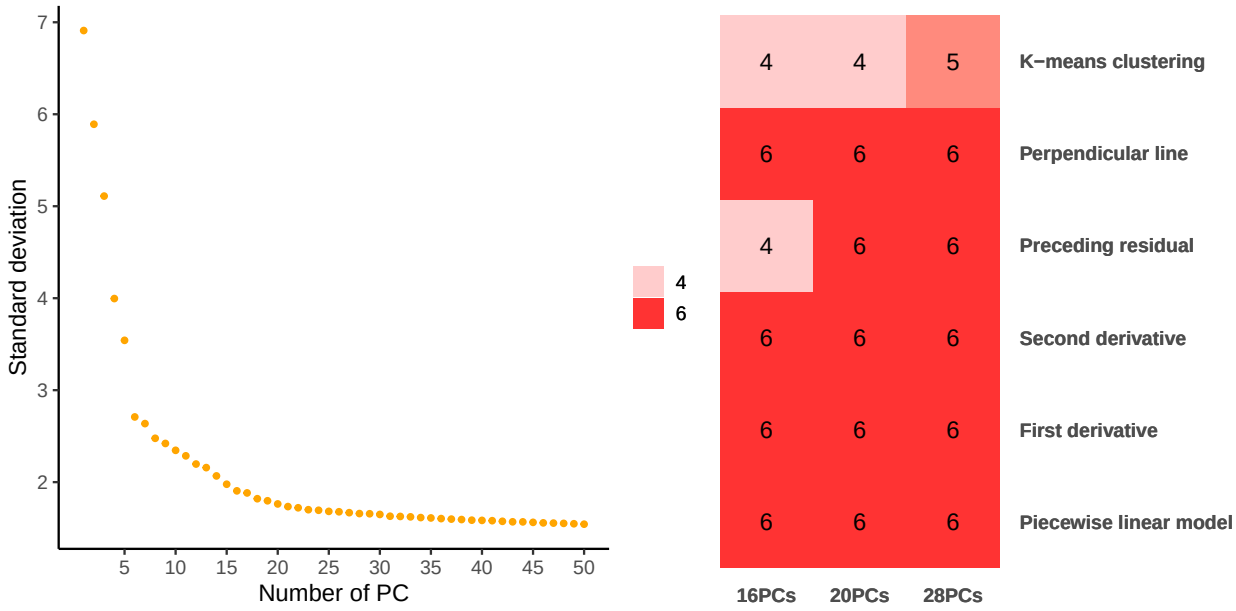
```
findPC(sdev = sdev,number = c(16,20,28),method = 'all',aggregate = 'voting')
```

```
## mode
##    6
```

Figure

The last argument 'figure' provides the option to print a heatmap showing the result.

```
findPC(sdev = sdev,number = c(16,20,28),method = 'all',figure = T)
```




```
##       Piecewise linear model First derivative Second derivative
## 16PCs                      6                6                 6
## 20PCs                      6                6                 6
## 28PCs                      6                6                 6
##       Preceding residual Perpendicular line K-means clustering
## 16PCs                  4                  6                  4
## 20PCs                  6                  6                  4
## 28PCs                  6                  6                  5
```